```
                          **********************
                          ***   TX REPORT   ***
                          **********************


        TRANSMISSION OK

        TX/RX NO                    4041
        CONNECTION TEL
        SUBADDRESS
        CONNECTION ID
        ST. TIME                    04/09 14:13
        USAGE T                     04'40
        PGS.                          3
        RESULT                      OK
```

United States Patent and Trademark Office

Technology Center 2100, Art Unit 2193 Alexandria, VA. 22313-1450

FAX COVER SHEET

Date : 4 / 9 / 07

Number of Pages (including cover) ___3___

To: Joseph Sawyer

101 6A 2,890

Phone : ( )

FAX: (650) 493-4549

From: Todd Ingberg

Phone : (571) 272-3723   DRAFT FAX (571) 273-3723

Official FAX: (571) 273-8300

Remark(s):

Final izing the agenda.

Summary + morkep

Agenda for After Final Interview
10/642,890

Participants
Charlie Bustamante A.K.A. – Munoz-Bustamante, Carlos

Exhibits for After Final Interview
Proposed amendment & Correspondence will be made of record

Topics
Overview of Invention – Briefly main inventive concepts over known prior art.
Proposed Amendment – How the proposed amendment clarifies the invention over prior art.

Comment
As usual the 30 minutes is mainly the Applicant's to make statements to add clarity.

Date and Time
Proposed date and time are confirmed if this agenda is accepted.

TODD INGBERG
PRIMARY EXAMINER

Applicant Initiated Interview Request Form

Application No.: 10/642,890	First Named Applicant: Neal R. Caliendo, Jr.
Examiner: Todd D. Ingberg	Art Unit: 2183 — Status of Application: Pending / Final

Tentative Participants:

(1) Examiner Todd D. Ingberg
(2) Joseph A. Sawyer, Jr. — *Attorney* ✓

(3) Charlie Bustamante — *Member Patent Bar*
(4) Rod Waiteman — *Inventor* ✓ R.W.

A.K.A → Munoz-Bustamante, Carlos (51, 549)

Proposed Date of Interview: April 11, 2007 — Proposed Time: 2:00PM est (AM/PM)

Type of Interview Requested:

(1) [×] Telephonic   (2) [ ] Personal   (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [X] YES   [ ] NO   *will be of Record*

If yes, provide brief description: *Proposed Amendment & correspondence*

Issues To Be Discussed

outline for Interview — 1 — 2

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) Rejections	___	X	[ ]	[ ]	[ ]
(2) ___	___	___	[ ]	[ ]	[ ]
(3) ___	___	___	[ ]	[ ]	[ ]
(4) ___	___	___	[ ]	[ ]	[ ]

[ ] Continuation Sheet Attached

Brief Description of Arguments to be Presented:

Draft Amendment As Forwarded to Examiner Ingberg / Prior Art objections

1. *Overview of Inventions - briefly main inventive concept over known Prior Art.*
2. *Proposed Amendment- How to clarify invention over prior art.*

An interview was conducted on the above-identified application on *DATE 4/11/07 2pm* — *Fine if agenda approved*

NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).

This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

/JOSEPH A. SAWYER, JR./
_____
Applicant/Applicant's Representative Signature

Examiner/SPE Signature

Joseph A. Sawyer, Jr.
_____
Typed/Printed Name of Applicant or Representative

30,801
_____
Registration Number, if applicable

Applicant Initiated Interview Request Form

Application No.: 10/642,890   First Named Applicant: Neal R. Caliendo, Jr.

Examiner: Todd D. Ingberg   Art Unit: 2183   Status of Application: Pending / Final

Tentative Participants:

(1) Examiner Todd D. Ingberg   (2) Joseph A. Sawyer, Jr.

(3) Charlie Bustamante   (4) Rod Walterman

Proposed Date of Interview: April 11, 2007   Proposed Time: 2:00PM est (AM/PM)

Type of Interview Requested:

(1) [x] Telephonic   (2) [ ] Personal   (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [ ] YES   [ ] NO

If yes, provide brief description:

Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) Rejections		X	[ ]	[ ]	[ ]
(2)			[ ]	[ ]	[ ]
(3)			[ ]	[ ]	[ ]
(4)			[ ]	[ ]	[ ]

[ ] Continuation Sheet Attached

Brief Description of Arguments to be Presented:

Draft Amendment As Forwarded to Examiner Ingberg / Prior Art objections

An interview was conducted on the above-identified application on _____

NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).

This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

/JOSEPH A. SAWYER, JR./

Applicant/Applicant's Representative Signature   Examiner/SPE Signature

Joseph A. Sawyer, Jr.

Typed/Printed Name of Applicant or Representative

30,801

Registration Number, if applicable

Applicant Initiated Interview Request Form

Application No.: 10/642,896   First Named Applicant: Neal R. Caliendo, Jr.

Examiner: Todd D. Ingberg   Art Unit: 2193   Status of Application: Pending / Final

Tentative Participants:

(1) Examiner Todd D. Ingberg   (2) Joseph A. Sawyer, Jr.

(3) Charlie Bustamante   (4) Rod Walterman

Proposed Date of Interview: Friday, April 6, 2007   Proposed Time: 1:30 PM   (AM/PM)

Type of Interview Requested:
(1) [×] Telephonic   (2) [ ] Personal   (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [×] YES   [ ] NO

If yes, provide brief description: Prior Art / Rejections

Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) Rejections	1-38	Dawson R. Engler	[ ]	[ ]	[ ]
(2)			[ ]	[ ]	[ ]
(3)			[ ]	[ ]	[ ]
(4)			[ ]	[ ]	[ ]

[ ] Continuation Sheet Attached

Brief Description of Arguments to be Presented:

Claims 1-12, 14-30, and 32-38 and newly presented 39-60.

An interview was conducted on the above-identified application on _____.

NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).

This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

/JOSEPH A. SAWYER, JR./
Applicant/Applicant's Representative Signature   Examiner/SPE Signature

Joseph A. Sawyer, Jr.
Typed/Printed Name of Applicant or Representative

30,801
Registration Number, if applicable

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Date: March ____, 2007
Neal R. Caliendo Jr.	Confirmation No.: 3348
Serial No.: 10/642,890	Group Art Unit: 2193
Filed: 08/18/2003	Examiner: Ingberg, Todd D.

For: METHOD FOR PROVIDING AN IMAGE OF SOFTWARE
INSTALLED ON A COMPUTER SYSTEM

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DRAFT
AMENDMENT

Sir:

In response to the Final Office Action dated March 3, 2007, please amend the above-identified application in the following manner:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Remarks begin on page 12 of this paper.

-1-

Amendments to the Claims

This listing of claims will replace all prior version, and listings, of claims in the application.

Listing of Claims:

1.   (Currently amended)   A method for providing an image of software installed on a computer system, the method comprising ~~the steps of:~~

(a)   deconstructing the image into at least one portion;

(b)   creating at least one module from the at least one portion of the image <u>utilizing</u> <u>information, wherein the information is selected from a group consisting of install information</u> <u>and uninstall information</u>; and

(c)   formatting the at least one module for use in a new image or at least a portion of a new image.

2.   (Currently amended)   The method of claim 1 wherein the deconstructing ~~step (a)~~ <u>the image</u> further comprises ~~the steps of:~~

(a2)   scanning the image; and

(a3)   identifying at least one portion of the image to be modularized.

3.   (Currently amended)   The method of claim 2 wherein the identifying ~~step (a3)~~ <u>at</u> <u>least one portion</u> comprises ~~the steps of:~~

(a3iii)   providing a list of portions of the image to be modularized; and

(a3iii) selecting at least one portion of the image to be modularized.

-2-

4.   (Original)   The method of claim 1 wherein the at least one portion of the image represents at least one software program.

5.   (Original)   The method of claim 4 wherein the at least one software program is hardware independent.

6.   (Original)   The method of claim 1 wherein the at least one portion of the image represents a plurality of software programs.

7.   (Original)   The method of claim 6 wherein the plurality of software programs comprises a combination of hardware-independent and hardware-dependent software programs.

8.   (Cancelled)

9.   (Currently amended)   The method of claim 1 wherein the creating step (b) at least one module further comprises the steps of:

(b2)   extracting the at least one portion of the image; and

(b3)   generating the at least one module from the extracted portion of the image.

10.   (Original)   The method of claim 9 wherein the extracted portion of the image comprises uninstall scripts.

11.   (Currently amended)   The method of claim 10 wherein the generating step (b3) at least one module comprises the steps of:

-3-

(b3ii)   scanning the uninstall scripts; and

(b3iii)   generating install scripts from the uninstall scripts.

12.   (Currently amended)   The method of claim 11 wherein the generating step (b3iii) install scripts comprises the steps of:

(b3iiiA)   reversing the order of the uninstall scripts;

(b3iiiB)   determining uninstall scripts from the uninstall scripts; and

(b3iiiC) configuring a portion of the install scripts.

13.   (Canceled)

14.   (Previously presented) The method of claim 1 wherein the software program is hardware-independent application software.

15.   (Original)   The method of claim 14 wherein the hardware-independent application software is a hardware-independent imaging tool.

16.   (Original)   The method of claim 1 wherein the module is hardware independent.

17.   (Currently amended)   The method of claim 1 wherein the creating step (b) at least one module further comprises the step of (b2) creating a plurality of modules from the at least one portion of the image.

18.   (Original)   The method of claim 17 wherein the plurality of modules comprises

-6-

a combination of hardware-independent and hardware-dependent modules.

19.   (Currently amended)   A computer-readable storage medium including a computer program for providing an image of software installed on a computer system, comprising instructions for:

(a)   deconstructing the image into at least one portion;

(b)   creating at least one module from the at least one portion of the image utilizing information, wherein the information is selected from a group consisting of install information and uninstall information; and

(c)   formatting the at least one module for use in a new image or at least a portion of a new image.

20.   (Currently amended)   The medium of claim 19 wherein the deconstructing instruction (a) the image further comprises the instructions of:

(a2)   scanning the image; and

(a3)   identifying the at least one portion of the image to be modularized.

21.   (Currently amended)   The medium of claim 20 wherein the identifying instruction (a3) at least one portion comprises the instructions of:

(a3ii) providing a list of portions of the image to be modularized; and

(a3iii) selecting the at least one portion of the image to be modularized.

22.   (Original)   The medium of claim 19 wherein the at least one portion of the image represents at least one software program.

-5-

23.　(Original)　The medium of claim 22 wherein the at least one software program is hardware independent.

24.　(Previously presented) The medium of claim 19 wherein the at least one portion of the image represents a plurality of software programs.

25.　(Previously presented) The medium of claim 24 wherein the plurality of software programs comprises a combination of hardware-independent and hardware-dependent software programs.

26.　(Cancelled)

27.　(Currently amended)　The medium of claim 19 wherein the creating ~~instruction (b)~~ at least one module further comprises ~~the instructions of:~~

　　~~(b2)~~ extracting the at least one portion of the image; and

　　~~(b3)~~ generating at least one module from the extracted portion of the image.

28.　(Original)　The medium of claim 27 wherein the extracted portion of the image comprises uninstall scripts.

29.　(Currently amended)　The medium of claim 28 wherein the generating ~~instruction~~ ~~(b3)~~ at least one module comprises the ~~instructions of:~~

　　~~(b3ii)~~ scanning the uninstall scripts; and

-6-

(b3iii) generating install scripts from the uninstall scripts.

30. (Currently amended)   The medium of claim 29 wherein the generating ~~instruction~~ ~~(b3iii)~~ install scripts comprises ~~the instructions of~~:

(b3iiiA)   reversing the order of the uninstall scripts;

(b3iiiB)   determining install scripts from the uninstall scripts; and

(b3iiiC)   configuring a portion of the install scripts.

31. (Cancelled)

32. (Previously presented) The medium of claim 19 wherein the software program is a hardware-independent application software.

33. (Original)   The medium of claim 32 wherein the hardware-independent application software is a hardware-independent imaging tool.

34. (Original)   The medium of claim 19 wherein the module is hardware independent.

35. (Currently amended)   The medium of claim 19 wherein the creating ~~instruction (b)~~ at least one module further comprises ~~the instruction of (b2)~~ creating a plurality of modules from the at least one portion of the image.

36. (Previously presented) The medium of claim 35 wherein the plurality of modules

comprises a combination of hardware-independent and hardware-dependent modules.

37.   (Cancelled)

38.   (Currently amended)   A computer-readable storage medium including a compute

program for providing an image of software installed on a computer system, comprising

instructions for:

(a)   deconstructing the image into at least one portion;

(b)   creating the at least one module from the at least one portion of the image utilizing

uninstall code; and

(c)   formatting the at least one module for use in at least a portion of a new image.

39.   (New) A system comprising:

a storage medium; and

a processing system coupled to the storage medium, the processing system including a

mechanism for deconstructing an image into at least one portion; creating at least one module

from at least one portion of the image utilizing information wherein the information is selected

from a group consisting of install information and uninstall information; and

formatting the at least one module for use in a new image or at least a portion of a new

image.

40.   (New) The system of claim 39 wherein the deconstructing the image comprises:

scanning the image; and

identifying at least one portion of the image to be modularized.

-8-

41.   (New) The system of claim 40 wherein the identifying at least one portion comprises:

providing a list of portions of the image to be modularized; and

selecting at least one portion of the image to be modularized.

42.   (New) The system of claim 39 wherein the at least one portion of the image represents at least one software program.

43.   (New) The system of claim 42 wherein the at least one software program is hardware independent.

44.   (New) The system of claim 39 wherein the at least one portion of the image represents a plurality of software programs.

45.   (New) The system of claim 44 wherein the plurality of software programs comprises a combination of hardware-independent and hardware-dependent software programs.

46.   (New) The system of claim 39 wherein the creating at least one module further comprises:

extracting the at least one portion of the image; and

generating the at least one module from the extracted portion of the image.

47.   (New) The system of claim 46 wherein the extracted portion of the image

comprises uninstall scripts.

48.   (New) The system of claim 47 wherein the generating at least one module comprises:

scanning the uninstall scripts; and

generating install scripts from the uninstall scripts.

49.   (New) The system of claim 48 wherein the generating install scripts comprises:

reversing the order of the uninstall scripts;

determining the uninstall scripts from the uninstall scripts; and

configuring a portion of the install scripts.

50.   (New) The system of claim 39 wherein the software program is hardware-independent application software.

51.   (New) The system of claim 50 wherein the hardware-independent application software is a hardware-independent imaging tool.

52.   (New) The system of claim 39 wherein the module is hardware independent.

53.   (New) The system of claim 39 wherein the creating at least one module further comprises creating a plurality of modules from the at least one portion of the image.

54. (New) The system of claim 54 wherein the plurality of modules comprises a combination of hardware-independent and hardware-dependent modules.

55. (New) A system comprising:

a storage medium; and

a processing system coupled to the storage medium, the processing system including a mechanism for deconstructing an image into at least one portion; creating at least one module from the at least one portion of the image utilizing uninstall code; and formatting the at least one module for use in a new image or at least a portion of a new image.

56. (New) The method of claim 1 wherein the at least one portion of the image comprises an operating system and code, wherein the code is selected from a group consisting of a set of drivers, a set of utilities and application software.

57. (New) The medium of claim 19 wherein the at least one portion of the image comprises an operating system and code, wherein the code is selected from a group consisting of a set of drivers, a set of utilities and application software.

58. (New) The medium of claim 38 wherein the at least one portion of the image comprises an operating system and code, wherein the code is selected from a group consisting of a set of drivers, a set of utilities and application software.

-11-

59.   (New) The system of claim 39 wherein the at least one portion of the image comprises an operating system and code, wherein the code is selected from a group consisting of a set of drivers, a set of utilities and application software.

60.   (New) The system of claim 55 wherein the at least one portion of the image comprises an operating system and code, wherein the code is selected from a group consisting of a set of drivers, a set of utilities and application software.

REMARKS

Claims 1-12, 14-30, and 32-38 are pending in the present application. Claims 1-3, 9, 11-12, 17, 19-21, 27, 29-30, 35 and 37-38 have been amended and no new matter has been added. Claims 8, 26, and 37 have been cancelled. Claims 13 and 31 have previously been cancelled. New claims 39-60 have been added. Accordingly, claims 1-7, 9-12, 14-25, 27, 30, 32-36 and 38-60 are now pending in the present application. Applicants find support for the claims generally throughout the specification, specifically on pages 5-14 and in Figures 1-5.

Present Invention

An improved process for providing an image of software installed on a computer system is disclosed. The process includes the steps of deconstructing an existing image and creating one or more modules from all or part of the image utilizing either install information or uninstall information. To deconstruct the image, the image is scanned to identify at least one portion of the image to be modularized. At least one portion of the image is then extracted, and at least one module is generated from that portion of the image. The module can then be formatted for use in a new image or part of a new image to be used with a particular software program, such as with a hardware-independent imaging tool or with other hardware-independent application software. An advantage of making an image modular is that it allows hardware-independent software programs (e.g., operating system, commonly used application software) to be abstracted or separated from hardware-dependent software programs (e.g., device drivers, hardware-dependent software). Modules can be added or removed from an image as needed, or can be combined to create new modular-based images.

-13-

Claim Rejections – 35 USC § 102

The Examiner states,

Claims 1-38 are rejected under 35 U.S.C. 102(b) as being anticipated by DERIVE: A Tool That Automatically Reverse-Engineers Instruction Encodings, Dawson R. Engler et al., ACM, 2000, pages 12-22.

DERIVE anticipates a method for providing an image of software installed on a computer system, the method comprising the steps of:

(a) deconstructing the image into at least one portion (Derive, Abstract, page 1, Reverse Engineering – installed software); and

(b) creating at least one module from the at least one portion of the image (Derive, Conclusion, page 19, instruction encoding and page 22, encoding structure, Figure 5 – emitter specification).

(c) formatting at least one module for use in a new image or at least a portion of a new image.

Examiner note: When taking the reference as a whole, please, look on page 14 Figure 1 at the information flow for a detailed view. DERIVE solver produces encoding description and the emitter generator feeds the instruction emitter, the presence of JIT is the Just in time Compiler which produces the new image in cooperation with the instruction emitter. Also, please look at the bottom of page 18 one of the last sentences on Linkers "...for only a few specific type of machine-dependent information, derived by feeding appropriate inputs to existing assemblies and linkers." The Linker by definition formats input into images. That is the role of the linker.

Applicants respectfully submit that the independent claims 1, 17 and 37-39 are not anticipated by the DERIVE reference. For ease of review, claim 1 is reproduced below:

1.   (Currently amended)   A method for providing an image of software installed on a computer system, the method comprising:
deconstructing the image into at least one portion, wherein the at least one portion of the image comprises an operating system and code, wherein the code is selected from a group consisting of a set of drivers, a set of utilities and application software;
creating at least one module from the at least one portion of the image utilizing information, wherein the information is selected from a group consisting of install information and uninstall information; and
formatting the at least one module for use in a new image or at least a portion of a new image.

Applicants submit that DERIVE discloses a method of reverse-engineering instruction encodings from pre-existing software (the system assembler) and uses the information extracted to construct dynamic linking libraries, object-level sandboxers, executable optimizers, and

-14-

linkers. Accordingly, DERIVE discloses reverse engineering instructions from software. In contrast, the present invention comprises deconstructing an image into at least one portion. As described in the specification, page 6, lines 11-13, an image comprises an operating system, at least one of a set of drivers, a set of utilities and/or application software. A portion of an image that is deconstructed from an image is clearly different than the reverse engineering of instructions as disclosed in DERIVE. In fact, DERIVE may be used as a portion of the recited invention by reverse engineering specific instructions but it is not capable of deconstructing an image into at least one portion.

In addition, a system in accordance with DERIVE reference copies a program from one instruction set to another instruction set. In contrast, in the present invention a module is created utilizing either install information or uninstall information from the at least one portion. Therefore, the creating of a module includes performing tasks required to install a new image on another computer system. For example, as stated in the specification at page 9, line 16-page 10, line 16,

> "There can be one or more portions and one or more modules generated from each portion depending on the specific application. In the preferred embodiment, the module is generated using uninstall code, also referred to as uninstall "scripts," which are commonly used to remove an installed software program. To generate the module from the uninstall scripts, the uninstall scripts are first scanned/searched and analyzed in reversed order to determine the actions that have taken place to install the software. The uninstall scripts are typically stored in an uninstall file, in a registry, or in the OS software and accessed from a dynamic-link library (DLL). The uninstall scripts typically include data such as application specific actions, decrement reference counts, shared DLL files, removed registry keys, pointers, links, files copied, and/or moved, etc.
>
> The module can then be installed onto a computer system or processed by an imaging tool by using install scripts that correspond to the uninstall scripts. The install scripts can be determined from information from the uninstall scripts in combination with log information related to the OS during an original installation. When a software program is installed under an OS, the OS maintains a log of actions taken during the installation process. For example, the log includes information on changes to the OS software. Such changes can include,

-15-

for example, newly shared DLLs reference counts, removed tags, decremented reference counts, etc. Such information can be used to configure the generated install scripts. The install scripts are ascertainable because the install and uninstall procedures are standardized. Accordingly, existing information in the image can be used in a reverse engineering process to create install scripts from the uninstall scripts. The install and uninstall scripts can be stored in a location specified by the user or in a default location such as with the files needed by related software programs."

As seen from the above, the invention as recited in the independent claims as well as the claims dependent thereon are not taught or suggested by the DERIVE reference because the DERIVE reference is directed to reverse engineering or copying of instructions to allow the instructions to be transferred from one instruction set to another instruction. As stated above, the recited invention provides at least a portion of an image which is clearly different from an instruction set. Furthermore, in the recited invention at least one module is created utilizing either uninstall information or install information from at least one portion of an image. DERIVE neither teaches nor suggests an equivalent process. Accordingly, this cooperation of elements are not taught, suggested or contemplated by the DERIVE reference.

Accordingly, claims 2-12, 14-18, 20-30, 32-36 and claims 40-54 are allowable since they depend from allowable base claims as well as for the above-stated reasons.

Claim Rejections – 35 USC § 103(a)

The Examiner states,

Claims 6, 7, 14-16, 19, 23, 26, 32-34 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over DERIVE in view of Modular Type-Based Reverse Engineering of Parameterized Types in Java Code, Dominic Duggan, ACM, 1999, pages 97-113.

Since, it is not clear if the independent the Applicant is claiming is from the input of the output of reverse engineering the Examiner has elected to reject the following claims under 35 U.S.C. 103(a).

Motivation to Combine DERIVE and JAVA

-16-

DERIVE teaches the emitting of C code (DERIVE, page 22). C code is not universally known to be platform independent. It is JAVA who teaches a well known platform independent language. Therefore, it would have been obvious to one of ordinary skill in the art to combine DERIVE and JAVA, because reverse engineering for a language like JAVA which is platform independent by the implementation of a virtual machine, would make a reverse engineering tool more flexible.

Applicant submits that the arguments hereinabove with respect to the DERIVE reference apply with equal force to this rejection since these claims depend from allowable base claims.

The JAVA reference describes a language independent platform but the combination of JAVA reference and the DERIVE reference provides for the reverse engineering of instructions utilizing a language independent platform. For the above-stated reasons, this is clearly different from the invention as recited in the above-identified claims.

Accordingly, claims 5, 7, 14-16, 18, 23, 25, 32-34 and 36 are allowable over the cited references either singly or in combination for the above-cited reasons in the above-identified claims.

New Claims

Applicants have added a new independent system claim 39 that has similar limitations to that in method and computer readable medium claims 1 and 19.

Accordingly, claim 39 is also allowable over the cited reference for the above-mentioned reasons. Furthermore, claims 40-54 are also allowable since they depend from an allowable base claim as well as for the above-stated reasons.

Independent computer readable medium claim 38 has been amended and new independent system 55 has been added to further define the scope and novelty of the present invention. Specifically both claims specifically recites "creating the at least one module from the at least one portion of the image utilizing uninstall code". Applicants respectfully submit

therefore that these claims are allowable for the same reasons as stated for independent claims 1 and 19.

New dependent claims 56-60 are added to further define the scope and novelty of the present invention.

-18-

Conclusion

For the above-identified reasons, Applicant respectfully requests reconsideration and allowance of claims 1-7, 9-12, 14-25, 27-30, 32-36, and 38-60 as now presented.

Applicant's attorney believes that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicant's attorney at the telephone number indicated below.

Respectfully submitted,

SAWYER LAW GROUP LLP

March ____, 2007 _____

_____

Joseph A. Sawyer, Jr.
Attorney for Applicant(s)
Reg. No. 30,801
(650) 493-4540